Reproducing Kernel Hilbert Spaces (1)

Manel Martínez-Ramón

ECE, UNM

October, 2018

- The algorithms presented in previous lessons are intended to find linear relationships between a set of features $\mathbf{x}_n$ and a variable (that we call label or regressor).

$$y_n = \mathbf{w}^\top \mathbf{x}_n + b + e_n$$

- However, in many situations, real relationships are rather nonlinear.
- The algorithms presented so far are then restrictive.

- Example: the relationship between the pressure applied to a quartz crystal and its voltage.
  - At low pressures, the relationship is linear.
  - At high pressures, the potential saturates.
- The relationship current – voltage in a diode is highly nonlinear.
- A classical way of nonlinear modelling is to use Volterra models. For a single dimension, a Volterra model can be written as

$$\hat{y}_n = \sum_{k=0}^{K} a_k x^k$$

We will present a related example below.

- Communications channel with the simplest FIR impulse response

$$h[n] = \delta[n] + a\delta[n-1]$$

Input: train of binary symbols $y[n] \in [+1, -1]$.

- Output of the channel:

$$x[n] = y[n] + ay[n-1] + g[n]$$

where $g[n]$ is additive gaussian noise.

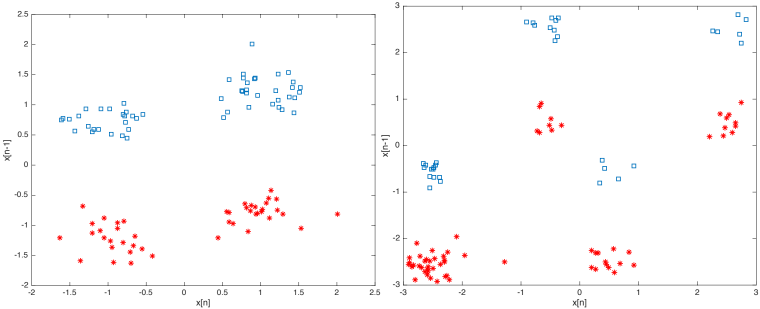
- We take a vector of two samples

$$\mathbf{x}_n = \{x[n], x[n-1]\}^\top$$

to determine the symbol corresponding to output $x[n]$.

Examples

- Representations of 100 data samples for $a = 0.2$ and $a = 1.5$



- A linear classifier cannot classify the data if $a = 1.5$.

Note: Of course, this academic example can be linearly solved using the Viterbi Algorithm, but this is another story.

What do we do?

Problem Assume that we only know about linear approaches.

Sol. Pass the data through a nonlinear transformation and then work linearly.

Classic approach: Volterra expansion.
Construct a nonlinear transformation with products between components. This are the components of a 3rd order transformation:

Order 0	1			
1st order	$x[n]$	$x[n-1]$		
2nd order	$x^2[n]$	$x^2[n-1]$	$x[n]x[n-1]$	
3rd order	$x^3[n]$	$x^3[n-1]$	$x^2[n]x[n-1]$	$x[n]x^2[n-1]$

Put all these components in a vector $\boldsymbol{\varphi}(\mathbf{x}_n) \in \mathbb{R}^{10}$. You just entered a space of 10 dimensions.

- Then you construct a linear estimator as

$$\hat{y}[n] = \mathbf{w}^\top \boldsymbol{\varphi}(\mathbf{x}_n)$$

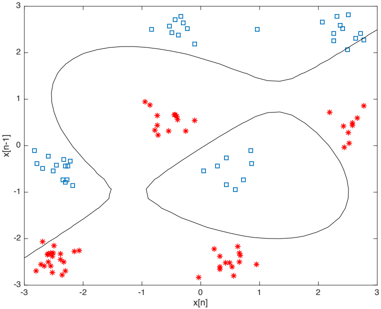
This function is linear in $\mathbf{w}$, but nonlinear with respect to $\mathbf{x}_n$

- You can adjust the parameters using a MMSE approach:

$$\mathbf{w} = (\boldsymbol{\Phi}\boldsymbol{\Phi}^\top)^{-1}\boldsymbol{\Phi}\mathbf{y}$$

- $\boldsymbol{\Phi}$ contains all values of $\boldsymbol{\varphi}(\mathbf{x}_n)$ and $\mathbf{y}$ a contains all bits $y[n]$. A train of bits (training sequence) is known by the receiver.

- Adjust the parameters following this algorithm and represent points $\mathbf{w}^\top \boldsymbol{\varphi}(\mathbf{x}_n) = 0$



- These points are the boundary that nonlinearly classify almost all points. Now you have to do it by yourself.

- Here we pass from of $\mathbb{R}^2$ to $\mathbb{R}^p$, $p = \left(\begin{array}{c} 2+3 \\ 3 \end{array} \right) = 10$:

$$1, x_1, x_2, x_1^2, x_2^2, x_1 x_2, x_1^2 x_2, x_1 x_2^2, x_1^3, x_2^3$$

  What if we need a higher order?

- In an input space of 2 dimensions, and with a Volterra expansion of order 5, we need 56 elements:

$$p = \left(\begin{array}{c} 2+5 \\ 5 \end{array} \right) = 56$$

- This is an example of the *the curse of dimensionality*.

Outcomes of this lesson

- We have seen an example of a simple problem that cannot be solved using a linear classifier.
- A nonlinear estimator can be constructed by a nonlinear transformation to a space of higher dimension.
- This solution suffers from the curse of dimensionality.